Pensieve header: February 3: Textbook (EIWL) chapters 13-18, unevaluated.

13. Arrays, or Lists of Lists

```
Table[x, 4]

Table[x, 4, 5]

Grid[Table[x, 4, 5]]

Grid[Table[RGBColor[r, 0, b], {r, 0, 1, .2}, {b, 0, 1, .2}]]

Grid[Table[i, {i, 4}, {j, 5}]]

Grid[Table[j, {i, 4}, {j, 5}]]

Grid[Table[i + j, {i, 5}, {j, 5}]]

ArrayPlot[Table[i * j, {i, 5}, {j, 5}]]

ArrayPlot[Table[RandomInteger[10], 30, 30]]

ArrayPlot[Table[RandomColor[], 30, 30]]

ImageData[Binarize[Rasterize["W"]]]

ArrayPlot[ImageData[Binarize[Rasterize["W"]]]]

1 - ImageData[Binarize[Rasterize["W"]]]

ArrayPlot[1 - ImageData[Binarize[Rasterize["W"]]]]
```

14. Coordinates and Graphics

```
ListPlot[{4, 3, 2, 1, 1, 1, 1, 2, 3, 4}]

ListLinePlot[{{1, 1}, {1, 5}, {6, 4}, {6, 2}, {2, 3}, {5, 5}}]

Table[RandomInteger[20], 10, 2]

RandomInteger[20, {10, 2}]

ListPlot[Table[RandomInteger[1000], 100, 2]]

Graphics[{Circle[{1, 1}], Circle[{1, 2}], Circle[{3, 1}]}]

Graphics[{Style[Circle[{1, 1}], Red],
  Style[Circle[{1, 2}], Green], Style[Circle[{3, 1}], Blue]}]

Graphics[Table[Circle[RandomInteger[50, 2]], 100]]

Graphics[Table[Circle[{x, y}], {x, 0, 10, 2}, {y, 0, 10, 2}]]

Graphics[{Circle[{1, 1}, 0.5], Circle[{1, 2}, 1.2], Circle[{3, 1}, 0.8]}]

Graphics[Table[Circle[{0, 0}, r], {r, 10}]]

Graphics[Table[Circle[{x, 0}, x], {x, 10}]]
```

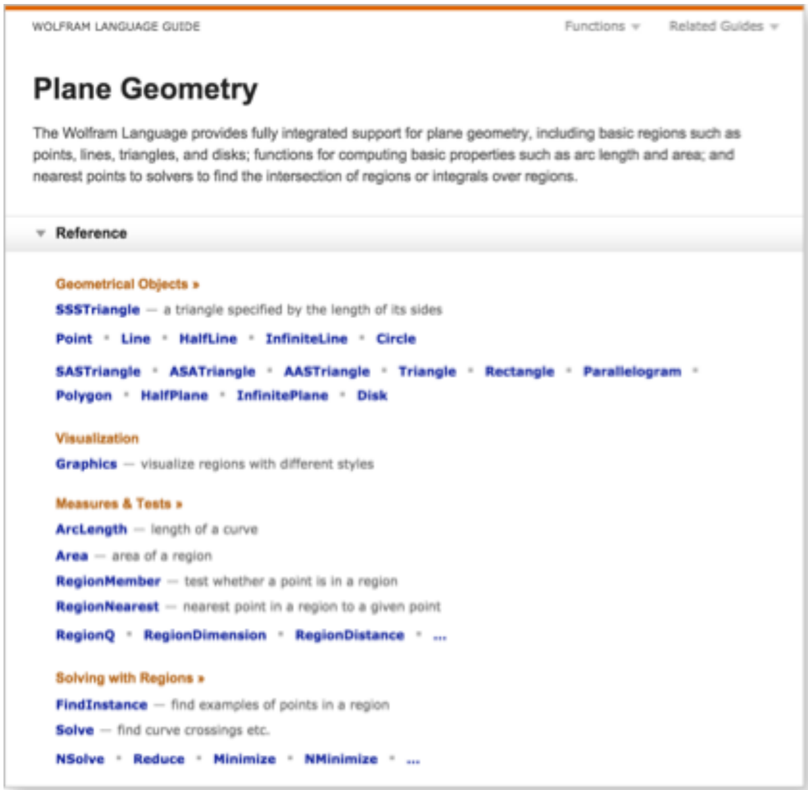
```
Graphics[Table[Circle[RandomInteger[50, 2], RandomInteger[10]], 100]]

Graphics[{RegularPolygon[{1, 1}, 1, 5], RegularPolygon[{3, 1}, 0.5, 7]}]

Graphics[{RegularPolygon[{1, 1}, 1, 5],
  Circle[{1, 1}, 1], RegularPolygon[{3, 1}, .5, 7], Disk[{2, 2}, .5]}]

Graphics[{Point[{0, 0}], Point[{2, 0}], Point[{1, 1.5}]}]

Graphics[Point[{{0, 0}, {2, 0}, {1, 1.5}}]]

Graphics[Line[{{0, 0}, {2, 0}, {1, 1.5}}]]

Graphics[Polygon[Table[RandomInteger[100], 20, 2]]]

Graphics3D[{Sphere[{0, 0, 0}], Sphere[{0, 0, 2}]}]

Graphics3D[Table[Sphere[{x, y, z}, 1/2], {x, 5}, {y, 5}, {z, 5}]]

Graphics3D[Table[Point[{x, y, z}], {x, 10}, {y, 10}, {z, 10}]]

Graphics3D[Table[Sphere[RandomInteger[10, 3]], 50]]

Graphics3D[Table[Style[Sphere[RandomInteger[10, 3]], Opacity[0.5]], 50]]

Manipulate[Graphics3D[{Sphere[{0, 0, 0}], Style[Sphere[{x, 0, 0}], Opacity[o]]}],
 {x, 1, 3}, {o, 0.5, 1}]
```

15. The Scope of the Wolfram Language

(aka RTFM)


Wolfram Language & System
Documentation Center
search documentation
Core Language & Structure
Data Manipulation & Analysis
Visualization & Graphics
Symbolic & Numeric Computation
Strings & Text
Graphs & Networks
Images
Geometry
Sound
Time-Related Computation
Geographic Data & Computation
Scientific and Medical Data & Computation
Engineering Data & Computation
Financial Data & Computation
Social, Cultural & Linguistic Data
Higher Mathematical Computation
Documents & Presentation
User Interface Construction
System Operation & Setup
External Interfaces & Connections
Cloud & Deployment
Fast Introduction for Programmers »
Wolfram|Alpha Knowledgebase Examples »
Legacy Documentation »


Images
Geometry
Sound
Geometric Computation Overview
Plane Geometry
Solid Geometry
Basic Geometric Regions
Mesh-Based Geometric Regions
Derived Geometric Regions
Geometric Properties
Geometric Solvers
Geometric Transformations
Importing & Exporting Geometry
Time-Related Computation
Engineering Data & Computation
Higher Mathematical Computation
System Operation & Setup
Scientific and Medical Data & Computation
Social, Cultural & Linguistic Data
User Interface Construction
Cloud & Deployment



WOLFRAM LANGUAGE GUIDE · Functions · Related Guides

Plane Geometry

The Wolfram Language provides fully integrated support for plane geometry, including basic regions such as points, lines, triangles, and disks; functions for computing basic properties such as arc length and area; and nearest points to solvers to find the intersection of regions or integrals over regions.

Reference

Geometrical Objects »

SSSTriangle — a triangle specified by the length of its sides

Point · **Line** · **HalfLine** · **InfiniteLine** · **Circle**

SASTriangle · **ASATriangle** · **AASTriangle** · **Triangle** · **Rectangle** · **Parallelogram** · **Polygon** · **HalfPlane** · **InfinitePlane** · **Disk**

Visualization

Graphics — visualize regions with different styles

Measures & Tests »

ArcLength — length of a curve

Area — area of a region

RegionMember — test whether a point is in a region

RegionNearest — nearest point in a region to a given point

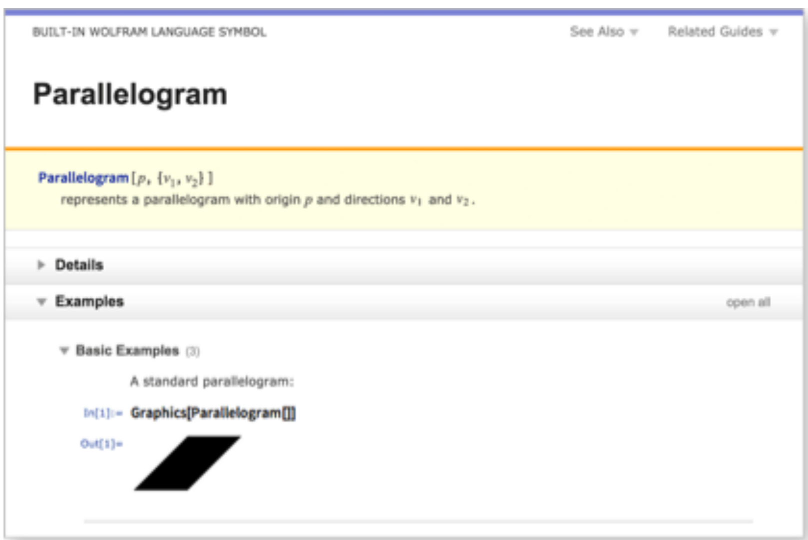
RegionQ · **RegionDimension** · **RegionDistance** · ...

Solving with Regions »

FindInstance — find examples of points in a region

Solve — find curve crossings etc.

NSolve · **Reduce** · **Minimize** · **NMinimize** · ...



BUILT-IN WOLFRAM LANGUAGE SYMBOL · See Also · Related Guides

Parallelogram

Parallelogram[p, {v_1, v_2}]
represents a parallelogram with origin p and directions v_1 and v_2.

Details

Examples · open all

Basic Examples (3)

A standard parallelogram:

In[1]:= Graphics[Parallelogram[]]

Out[1]=

```
Graphics[Parallelogram[]]
```

16. Real- World Data

```
c = Canada (country)
```

```
c // InputForm
```

```
cf = c@"Flag"
```

```
ColorNegate[cf]
```

```
EntityValue[ Canada (country) , flag ]
```

For the rest, RTFM. Yet,

```
Russia (country) ["BorderingCountries"]
```

```
all countries, dependencies, and territories (countries)
```

```
V = EntityList[ all countries, dependencies, and territories (countries) ]
```

```
#["BorderingCountries"] & /@ V
```

```
V = CountryData["Countries"]
```

```
Table[CountryData[c, "BorderingCountries"], {c, V}]
```

```
Graph[{1, 2}, {1 ⟷ 2}]
```

```
Ed = Union[Flatten[Table[
    Table[c ⟷ c1, {c1, CountryData[c, "BorderingCountries"]}],
    {c, V}]]]
```

```
G = Graph[V, Ed]
```

17. Units

```
Sin[30 °]
```

```
UnitConvert[ C$ , $ ]
```

For the rest, RTFM.

18. Geocomputation

```
Here // GeoGraphics
```

For the rest, RTFM.